

MIMA YOSHIAKI

DERWENT-ACC-NO: 1999-317783  
DERWENT-WEEK: 199927  
COPYRIGHT 1999 DERWENT INFORMATION LTD

TITLE: Print data transfer method in parallel processor network - involves classifying and defining data corresponding to user's printing demand specification, in protocol followed by transmitting protocol to respective printer

PATENT-ASSIGNEE: IBM CORP[IBMC]

PRIORITY-DATA: 1997JP-0248440 (September 12, 1997)

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
JP 11110143 A	April 23, 1999	N/A	016	G06F 003/12

APPLICATION-DATA:

PUB-NO	APPL-DESCRIPTOR	APPL-NO	APPL-DATE
JP11110143A	N/A	1997JP-0248440	September 12, 1997

INT-CL\_(IPC): B41J029/38; G06F003/12 ; G06F015/16

ABSTRACTED-PUB-NO: JP11110143A

BASIC-ABSTRACT: NOVELTY - The situation of printer resources are investigated based on the user's printing demand specification. The data corresponding to the specification are classified and defined in a protocol which is then transmitted to the corresponding printer.

USE - For transferring print data in parallel processor network.

ADVANTAGE - Offers parallel printing operation of several printing jobs simultaneously using moving agent technique. The procedure of choosing another printer automatically, in case of failure in a particular printer is offered, since several printers are connected to the network. ☆

CHOSEN-DRAWING: Dwg.1/9

TITLE-TERMS:

PRINT DATA TRANSFER METHOD PARALLEL PROCESSOR NETWORK

CLASSIFY DEFINE DATA  
CORRESPOND USER PRINT DEMAND SPECIFICATION PROTOCOL  
FOLLOW TRANSMIT PROTOCOL  
RESPECTIVE PRINT

DERWENT-CLASS: P75 T01

EPI-CODES: T01-C05A; T01-M02;

SECONDARY-ACC-NO:

Non-CPI Secondary Accession Numbers: N1999-238005

Publication Number	Application Number	Doc Kind	Pages	Country
11110143	24844097	A	16	JP
Publication Date	Application Date	Emperor Code		
Apr 23, 1999	Sep 12, 1997	H		
Priority Number	Inventor			
24844097	MIMA YOSHIKI			

OK	Print Detail
----	--------------

Int'l Classification	Patent/Applicant
G06F3/12	IBM

Title Of Invention
PRINTER SYSTEM EXECUTE PRINTING, PRINTING EXECUTING METHOD, COMPUTER AND PRINTER SYSTEM

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-110143

(43) 公開日 平成11年(1999) 4月23日

(51) Int.Cl.<sup>9</sup>

識別記号

F I

G 0 6 F 3/12

G 0 6 F 3/12

A

D

B 4 1 J 29/38

B 4 1 J 29/38

Z

G 0 6 F 15/16

3 8 0

G 0 6 F 15/16

3 8 0 D

審査請求 未請求 請求項の数27 O L (全 16 頁)

(21) 出願番号

特願平9-248440

(22) 出願日

平成9年(1997) 9月12日

(71) 出願人 390009531

インターナショナル・ビジネス・マシーンズ・コーポレーション

INTERNATIONAL BUSINESS MACHINES CORPORATION

アメリカ合衆国10504、ニューヨーク州

アーモンク (番地なし)

(72) 発明者 美馬 鏡亮

神奈川県大和市下鶴間1623番地14 日本アイ・ピー・エム株式会社 東京基礎研究所内

(74) 代理人 弁理士 坂口 博 (外1名)

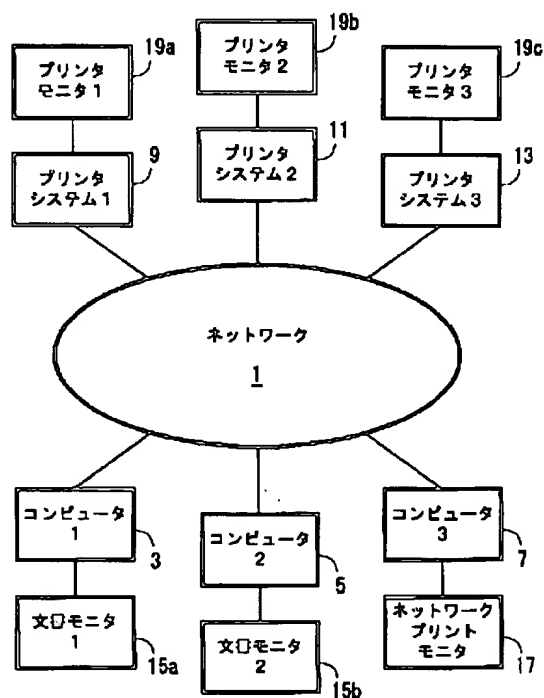
最終頁に続く

(54) 【発明の名称】 印刷をプリンタ・システムに実行させる方法、印刷実行方法、コンピュータ、及びプリンタ・システム

(57) 【要約】 (修正有)

【課題】 ネットワークに複数のプリンタが接続されている環境で、複数のプリンタを組み合わせて並列印刷可能とする方法を提供する。

【解決の手段】 文書ごとに文書モニタというプロセスが動作して、現在利用可能なプリンタ資源の状況を調査し、印刷の要求を最適化するために、印刷する文書を複数の印刷ジョブとして分割する。これらの印刷ジョブへの分割は、ターンアラウンド時間の減少などさまざまな最適化要求に対応することが可能である。印刷ジョブは、「xページからyページまで」のようにページ単位での印刷を管理するプログラムであり、文書印刷の実行をモニタする機能をもつ。このプログラムにより、印刷ジョブごとに異なる挙動を行うことが可能になる。印刷ジョブ・エージェントがプリンタ・モニタに到着すると、その印刷ジョブ・エージェントに定義されたプログラムがプリンタ・モニタによって実行状態にされる。



## 【特許請求の範囲】

【請求項1】複数のプリンタ・システムとコンピュータとが接続されたネットワークにおいて、前記コンピュータが印刷を前記プリンタ・システムに実行させる方法であって、

印刷データと、当該印刷データの印刷制御データとを受け取るステップと、

前記複数のプリンタ・システムに関する情報を取得する検査ステップと、

前記複数のプリンタ・システムに関する情報と前記印刷制御データとを参照して、前記印刷データをどのプリンタ・システムにどのように割り振るか決定し、決定された割り振りに従い前記印刷データを分割する分割ステップと、

分割された印刷データの各々について、当該分割された印刷データと印刷コントローラを含む移動エージェントを作成するステップと、

前記移動エージェントを当該移動エージェントが含む印刷データが割り振られた前記プリンタ・システムに送信するステップと、を含む方法。

【請求項2】前記印刷制御データは、ユーザの印刷要求仕様と、出力態様とを含む請求項1記載の方法。

【請求項3】前記移動エージェントを作成するステップが、前記出力態様を参照して、前記印刷コントローラの設定データを作成するステップをさらに含む請求項2記載の方法。

【請求項4】前記印刷コントローラが、前記プリンタ・システムの状況を監視するモニタと、前記モニタが通知する前記プリンタ・システムのエラーに対応するエラー・ハンドラと、前記印刷データの出力を制御する制御モジュールと、を含む請求項1記載の方法。

【請求項5】印刷終了の通知を受信するステップと、全印刷の終了を確認したことに応答して、印刷レポートを作成するステップと、をさらに含む請求項1記載の方法。

【請求項6】前記プリンタ・システムのエラー内容及び未印刷データを含む移動エージェントを受信するステップと、前記未印刷データに対して、前記分割ステップ以下を実行するステップと、をさらに含む請求項1記載の方法。

【請求項7】前記検査ステップが、各プリンタ・システムの性能を保持し、各プリンタ・システムの状況をモニタするネットワーク・プリント・モニタから、前記複数のプリンタ・システムに関する情報を読み出すステップを含む請求項1記載の方法。

【請求項8】各々移動エージェントの実行環境を含むア

リント・システムと、コンピュータとが接続されたネットワークにおいて、前記プリンタ・システムで印刷を実行する方法であって、

印刷データと印刷コントローラとを含む移動エージェントを受信するステップと、

前記移動エージェントを実行可能状態に変換するステップと、

前記印刷コントローラがそれ自身の設定に従って前記印刷データを出力するステップと、

10 全ての印刷データを印刷し終えた場合、印刷終了を前記移動エージェント送信元に通知するステップと、を含む印刷実行方法。

【請求項9】前記プリンタ・システムにエラーが発生した場合、当該エラーの内容を前記印刷コントローラが検査するステップと、

前記エラーの内容が所定の条件を満たしている場合、前記エラーの内容及び未印刷データを含む前記移動エージェントを、送信元に送信するステップと、をさらに含む請求項8記載の印刷実行方法。

20 【請求項10】印刷データ及び印刷コントローラを含む第2移動エージェントを受信するステップと、

前記移動エージェントによる印刷処理が終了しているかどうか判断するステップと、

終了していない場合、前記第2移動エージェントを実行不能状態で保管するステップと、をさらに含む請求項8記載の印刷実行方法。

【請求項11】前記移動エージェントが実行可能状態でなくなったことに応答して、前記第2エージェントを実行可能状態に変換するステップと、

30 をさらに含む請求項10記載の印刷実行方法。

【請求項12】ネットワークに接続された複数のプリンタ・システムに印刷を実行させるコンピュータであって、

前記複数のプリンタ・システムに関する情報を取得する検査モジュールと、

前記複数のプリンタ・システムに関する情報と入力される印刷制御データとを参照して、印刷データをどのプリンタ・システムにどのように割り振るか決定し、決定された割り振りに従い前記印刷データを分割する分割モジュールと、

分割された印刷データの各々について、当該分割された印刷データと印刷コントローラを含む移動エージェントを作成するモジュールと、

前記移動エージェントを当該移動エージェントが含む印刷データが割り振られた前記プリンタ・システムに送信するモジュールと、を有するコンピュータ。

【請求項13】前記印刷制御データは、ユーザの印刷要求仕様と、出力態様とを含む請求項12記載のコンピュータ。

【請求項14】前記移動エージェントを作成するモジュールが、

前記出力態様を参照して、前記印刷コントローラの設定データを作成する手段をさらに含む請求項13記載のコンピュータ。

【請求項15】前記印刷コントローラが、  
前記プリンタ・システムの状況を監視するモニタと、  
前記モニタが通知する前記プリンタ・システムのエラーに対応するエラー・ハンドラと、  
前記印刷データの出力を制御する制御モジュールと、  
を含む請求項12記載のコンピュータ。

【請求項16】印刷終了の通知を受信するモジュールと、  
全印刷の終了を確認したことに応答して、印刷レポートを作成するモジュールと、  
をさらに含む請求項12記載のコンピュータ。

【請求項17】前記プリンタ・システムのエラー内容及び未印刷データを含む移動エージェントを受信するモジュールと、  
前記未印刷データを前記移動エージェントから取り出し、前記分割モジュールに入力するモジュールと、  
をさらに含む請求項12記載のコンピュータ。

【請求項18】前記検査モジュールが、  
各プリンタ・システムの性能を保持し、各プリンタ・システムの状況をモニタするネットワーク・プリント・モニタから、前記複数のプリンタ・システムに関する情報を読み出す手段を含む請求項12記載のコンピュータ。

【請求項19】ネットワークに接続されるプリンタ・システムであって、  
移動エージェントの実行環境であるプリンタ・モニタを有し、  
前記プリンタ・モニタは、  
印刷データと印刷コントローラとを含む移動エージェントを実行可能状態に変換する変換モジュールと、  
2以上の移動エージェントを受信した場合、実行可能状態に変換された移動エージェントが印刷処理を終了するまで他の移動エージェントを保管し、当該終了後他の1の移動エージェントを前記変換モジュールに渡すモジュールと、  
を含むプリンタ・システム。

【請求項20】前記プリンタ・モニタは、前記プリンタ・システムの状態を監視し、前記印刷コントローラに状態を伝えるモニタ・モジュールをさらに含む請求項20記載のプリンタ・システム。

【請求項21】前記モニタ・モジュールは、  
前記ネットワークに設けられ且つ前記ネットワークに接続されたプリンタ・システムに関する情報を収集するネットワーク・プリント・モニタに、当該モニタ・モジュールが存在するプリンタ・システムの状況を通知することを特徴とする請求項20記載のプリンタ・システム。

【請求項22】ローカル・エリア・ネットワークであって、

少なくとも1のコンピュータと、  
複数のプリンタ・システムと、  
を有し、  
前記コンピュータは、  
前記複数のプリンタ・システムに関する情報を取得する検査モジュールと、  
前記複数のプリンタ・システムに関する情報と入力される印刷制御データとを参照して、印刷データをどのプリンタ・システムにどのように割り振るか決定し、決定された割り振りに従い前記印刷データを分割する分割モジュールと、

分割された印刷データの各々について、当該分割された印刷データと印刷コントローラを含む移動エージェントを作成するモジュールと、  
前記移動エージェントを当該移動エージェントが含む印刷データが割り振られた前記プリンタ・システムに送信するモジュールと、

を有し、  
各前記プリンタ・システムは、  
前記移動エージェントを受信する受信モジュールと、  
前記移動エージェントを実行可能状態に変換する変換モジュールと、  
前記受信モジュールが2以上の移動エージェントを受信した場合、実行可能状態に変換された移動エージェントが存在しなくなるまで他の移動エージェントを保管し、当該存在しなくなることに応答して他の1の移動エージェントを前記変換モジュールに渡すモジュールと、

を有するローカル・エリア・ネットワーク。  
【請求項23】複数のプリンタ・システムとコンピュータとが接続されたネットワークにおいて、印刷を前記プリンタ・システムに実行させる移動エージェントを前記コンピュータに作成させるプログラムを格納した記憶媒体であって、  
前記プログラムは、  
印刷データと、当該印刷データの印刷制御データとを受け取るステップと、  
前記複数のプリンタ・システムに関する情報を取得する検査ステップと、  
前記複数のプリンタ・システムに関する情報と前記印刷制御データとを参照して、前記印刷データをどのプリンタ・システムにどのように割り振るか決定し、決定された割り振りに従い前記印刷データを分割する分割ステップと、  
分割された印刷データの各々について、当該分割された印刷データと印刷コントローラを含む移動エージェントを作成するステップと、  
を含む、記憶媒体。

【請求項24】前記プログラムが、

前記移動エージェントを受信する受信モジュールと、  
前記移動エージェントを実行可能状態に変換する変換モジュールと、  
前記受信モジュールが2以上の移動エージェントを受信した場合、実行可能状態に変換された移動エージェントが存在しなくなるまで他の移動エージェントを保管し、当該存在しなくなることに応答して他の1の移動エージェントを前記変換モジュールに渡すモジュールと、  
を有するローカル・エリア・ネットワーク。

【請求項25】複数のプリンタ・システムとコンピュータとが接続されたネットワークにおいて、印刷を前記プリンタ・システムに実行させる移動エージェントを前記コンピュータに作成させるプログラムを格納した記憶媒体であって、  
前記プログラムは、  
印刷データと、当該印刷データの印刷制御データとを受け取るステップと、  
前記複数のプリンタ・システムに関する情報を取得する検査ステップと、  
前記複数のプリンタ・システムに関する情報と前記印刷制御データとを参照して、前記印刷データをどのプリンタ・システムにどのように割り振るか決定し、決定された割り振りに従い前記印刷データを分割する分割ステップと、  
分割された印刷データの各々について、当該分割された印刷データと印刷コントローラを含む移動エージェントを作成するステップと、  
を含む、記憶媒体。

【請求項26】前記プログラムが、

全印刷の終了を確認したことに応答して、印刷レポートを作成するステップ、

をさらに含む、請求項23記載の記憶媒体。

【請求項25】印刷データと、

前記印刷データを印刷するプリンタ・システムの状況を監視するモニタと、

前記モニタが通知する前記プリンタ・システムのエラーに対応するエラー・ハンドラと、

前記印刷データの出力を制御する制御モジュールと、

を有する移動エージェントを格納した記憶媒体。

【請求項26】印刷用デバイス・ドライバを格納した記憶デバイスであって、

前記印刷用デバイス・ドライバは、

印刷データと、当該印刷データの印刷制御データとを受け取るステップと、前記複数のプリンタ・システムに関する情報を取得する検査ステップと、前記複数のプリンタ・システムに関する情報と前記印刷制御データとを参照して、前記印刷データをどのプリンタ・システムにどのように割り振るか決定し、決定された割り振りに従い

前記印刷データを分割する分割ステップと、分割された印刷データの各々について、当該分割された印刷データと印刷コントローラを含む移動エージェントを作成するステップとを実行するプログラムを起動するステップと、

前記プログラムに印刷データ及び印刷制御データを出力するステップとを含む、記憶デバイス。

【請求項27】複数のプリンタ・システムがネットワークに接続されたシステムであって、

ある文書の各ページを前記複数のプリンタ・システムのいずれかに割り当て、前記複数のプリンタ・システムの各々で前記文書の異なるページを並列に印刷するシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、印刷処理に関し、より詳しくは、移動エージェント技術の印刷処理への適用に関する。

【0002】

【従来の技術】コンピュータとプリンタをネットワークで結び、1又は複数のコンピュータが1又は複数のプリンタを共有することが可能である。このようなネットワークでは、プリンタにIDを割り当て、コンピュータの側からこのIDを用いてプリンタに印刷要求を送るようになっている。プリンタは、コンピュータからの要求（印刷ジョブ）を一時的に記憶し、プリンタの機械装置側で印刷が可能になり次第、次々に印刷ジョブの実行を行うスプール機能を有するのが一般的である。この機能により、複数の文書の印刷を行うときプリンタのアイドル時間を最小限にして印刷することが可能となる。

【0003】以上のシステムでは、印刷要求を実行した

場合、印刷開始時まで印刷ジョブが特定のプリンタに割り当てられる。この割り当ての変更はユーザ操作によってのみ可能である。プリンタでは、電子的な手段で制御不可能な紙詰まりや紙切れ、インクやトナー切れ等のエラーが頻繁に発生し得る。このような障害が発生した場合、印刷システムは作業を中断するのみであり、別のプリンタに出力を行うといったことを行う機能を提供するシステムは存在しない。このため、これらの障害がオペレータによって解消されるまで、障害が発生したプリンタに割り当てられた印刷ジョブは、障害回復を待つことになる。

【0004】また、大量のデータからなる1つ又は複数の文書を出力する場合、複数のプリンタを同時に利用すれば全体の処理時間を削減することができるが、現在の印刷制御装置では自動的にこのような並列印刷を行うことは行われていない。現在のシステムでこのような並列印刷を実行するためには、アプリケーション・プログラムやユーザの側から明示的に印刷ジョブを個別のプリンタに割り当てることが必要となり、ユーザ又はアプリケーション・プログラムに負担がかかる。

【0005】

【発明が解決しようとする課題】複数のプリンタを制御することができる中規模以上のコンピュータでは、複数のプリンタの中で負荷の低いプリンタに新たに発生した印刷ジョブを割り当てることにより、プリンタの稼働率を高めることができる。このような制御方法は、高速且つ信頼性の高いプリンタでは大きな効果を有する。

【0006】しかし、昨今のオフィス環境では、個人使用を目的とする小規模、低速且つ信頼性の低い印刷装置が複数使用されている。このような環境では、印刷ジョブの単位でプリンタを割り当てて印刷することは必ずしも最善とは言えない。今までの印刷制御方法では、一つの印刷ジョブが特定のプリンタを占有する時間が長い反面、使用されていない、ネットワークに接続されたプリンタが多く存在するという資源の有効利用に反した状況を生じている。

【0007】よって、本発明の目的は、ネットワークに複数のプリンタが接続されている環境で、あるプリンタに障害が発生しても、自動的に別のプリンタを選択しそのプリンタで印刷を実行できるようにする方法を提供することである。

【0008】また、ネットワークに複数のプリンタが接続されている環境で、複数のプリンタを組み合わせで並列印刷可能とする方法を提供することも目的である。

【0009】さらに、移動エージェントの技術を用いて、印刷処理を実施する方法を提供することも目的である。

【0010】

【課題を解決するための手段】本発明は、文書を印刷する必要が生じ、印刷の要求をプリンタに出力する時、文

書ごとに文書モニタというプロセスを実行する。文書モニタは、現在利用可能なプリンタ資源の状況を調査し、印刷の要求を最適化するために、印刷する文書を複数の印刷ジョブとして分割する。これらの印刷ジョブへの分割は、ターンアラウンド時間の減少、印刷スループットの向上、印刷品質の最適化、障害回避などさまざまな最適化要求に対応することが可能である。印刷ジョブは、ある文書の「xページからyページまで」のようにページ単位での印刷を管理するプログラムであり、文書印刷の実行をモニタする機能をもつ。このプログラムにより、印刷ジョブごとに異なる挙動を行うことが可能になる。

【0011】具体的手順は以下になる。一台のプリンタに対して対応するプリンタ・モニタと呼ぶ移動エージェントの実行環境がプリンタ上または関連づけられたコンピュータ上で実行される。このプリンタ・モニタはプリンタと接続されており、プリンタとの通信手段を介して、プリンタに印刷用のデータやコマンドを送ったり、プリンタの状態を調べたりすることが可能である。プリンタ・モニタは定期的にプリンタの状態を調べることにより、それが接続されたプリンタの状況を知り、ネットワークを介して他のプロセスに伝える作業が可能である。この機能によりネットワーク上のコンピュータは、「現在、どのプリンタが紙詰まりを起こしているか」や、「どのプリンタはあと何枚プリント作業が残っているか」といったような状態を常にモニタすることが可能になる。なお、本願発明の移動エージェントは、プログラムとデータとが一体となっており、ネットワーク内を移動可能なオブジェクトという意味で用いる。

【0012】プリンタ・モニタは印刷ジョブ・エージェント（移動エージェント）を受け取ることができる。印刷ジョブ・エージェントがプリンタ・モニタに到着すると、その印刷ジョブ・エージェントに定義されたプログラムがプリンタ・モニタによって実行状態にされる。プリンタ・モニタは複数の印刷ジョブ・エージェントを受け取ることができるが、複数の印刷ジョブ・エージェントのうちプリンタに印刷要求をすることが許されるのは、ただ一つの印刷ジョブ・エージェントだけである。実行状態でかつ印刷可能な状態に置かれた印刷ジョブ・エージェントはプリンタに印刷要求を出すとともに、プリンタ・モニタを介して、または直接プリンタと通信することによってプリンタの状態を常時監視する。

【0013】印刷ジョブ・エージェントのプログラムは、プリンタからのエラーが生じなければ、定められた文書の印刷を行い、正常にプリントが完了したことを要求元に通知する。作業中にプリンタ上でエラーが生じたときは印刷ジョブ・エージェント中に定義されたプログラムがそのエラーの内容を判断し、必要に応じて印刷要求元に状況を通知する。通知は、メッセージを発してもよいし、印刷データ及び印刷制御データをもって自ら印刷要求元に戻ることも可能である。自ら印刷要求元に戻

る際には、未印刷データを保持するして戻るようにしてもよい。

【0014】ネットワーク・プリント・モニタはネットワーク上の計算機上におかれ、ネットワーク上におけるグローバルな情報の管理を行うために作られる。プリンタ・モニタからの報告もしくはこのネットワーク・プリント・モニタからの問い合わせによりプリンタの状況を確認し、プリンタの性能や状態をモニタするプロセスである。複数のプリンタの状況を記録するための表と、プリンタからの報告を受けるための機構及びプリンタの状況を定期的にチェックする機構を持っている。

【0015】文書モニタは、文書を印刷するときに生成される印刷データをもったプロセスである。特に、文書を複数の印刷ジョブに分割して印刷する場合に作られる場合に有効なしくみである。このプロセスでは印刷することを要求されたページをどのように分割し、どのプリンタに作業を割り当てたか、実行中の印刷作業がどのような状態にあるのか、どのプリンタがどれぐらいの待ち時間で利用できるか、またカラー印刷ができるか、どれぐらいの処理能力があるかなどといった印刷のための資源の管理を行う。文書モニタは印刷するためのデータの情報ならびに、プリンタ単位で生成した印刷ジョブの表を持っている。同時に、印刷が終了したとき、どのページがどのプリンタで出力されているかなどを管理し報告する機能をもつ。このプロセスは必要に応じて生成された文書印刷とそのレポートを作成したあとは消滅するようにすることも可能である。

【0016】印刷を行うにあたっては、どのような制限を設けて印刷をするのかに関するオプションをユーザが指定し、これにしたがってプリンタ・ドライバは、文書モニタを生成する。このようにして生成された文書モニタは一又は複数の印刷ジョブ・エージェントを生成する。文書モニタが生成した印刷ジョブ・エージェントは、文書モニタによって実行時において印刷実行に都合の良いプリンタを割り当てられ、それぞれのプリンタで印刷が行われる。印刷が無事に終わったときは単純に文書が出力されたプリンタ名のリストが返される。エラーが生じたときは文書モニタによって出力候補となるプリンタが再指定され、まだ印刷されていないと推測される文書の印刷についての割り当てが行われる。

【0017】以上を再度まとめると、複数のプリンタ・システムとコンピュータとが接続されたネットワークにおいて、コンピュータが印刷をプリンタ・システムに実行させる際には、まず印刷データと、当該印刷データの印刷制御データとを受け取る。そして、複数のプリンタ・システムに関する情報（例えば、性能及び状況）を取得する。次に、この情報（性能及び状況）と印刷制御データとを参照して、印刷データをどのプリンタ・システムにどのように割り振るか決定し、決定された割り振りに従い印刷データを分割する。その後、分割された印刷



データの各々について、当該分割された印刷データと印刷コントローラを含む移動エージェントを作成する。この移動エージェントを当該移動エージェントを含む印刷データが割り振られたプリンタ・システムに送信する。このようにすれば、印刷ジョブの分散及び並列処理が実施可能となる。なおプリンタ・システムは、コンピュータとプリンタの組み合わせや、ネットワーク対応でコンピュータが内蔵されたプリンタを含むことを意味する。

【0018】上記印刷制御データが、ユーザの印刷要求仕様と、出力態様とを含むようにすることが考えられる。例えば、ユーザの印刷要求仕様とはカラー印刷／モノクロ印刷や高速印刷主体であるといったデータであり、出力態様とは、例えばページ1, 2, 3, 1, 2, 3の順番で印刷するか、1, 1, 2, 2, 3, 3で印刷するか等のデータである。

【0019】上記の移動エージェントを作成する処理では、この出力態様を参照して、印刷コントローラの設定データを作成するようにすることも考えられる。上記のような出力順番を設定データとして印刷コントローラに設定し、移動エージェントがプリンタ・システムの実行環境で設定データに従い、印刷データを出力するようにする。

【0020】この印刷コントローラは、プリンタ・システムの状況を監視するモニタと、モニタが通知するプリンタ・システムのエラーに対応するエラー・ハンドラと、印刷データの出力を制御する制御モジュールとを含むようにすることができる。エラー・ハンドラを含むことによって、プリンタ・システムにおいて頻繁に起こる障害に自動的に対応できるようになる。

【0021】印刷終了の通知を受信し、全印刷の終了を確認したことに応答して、印刷レポートを作成するように構成することも可能である。電子メールやメッセージをディスプレイに表示する等の手段で、ユーザに印刷終了を通知し、どのプリンタにどのページが印刷されたかを知らせることにより、ユーザが簡単に印刷物を回収できるようにする。なお、障害が発生した場合に印刷レポートを作成するようにすることも可能である。

【0022】プリンタ・システムのエラー及び未印刷データを含む移動エージェントを受信した場合には、未印刷データに対して、分割処理以下を実行ようにする。障害が発生した時点で最良の印刷ジョブ割り当てを実施するためである。

【0023】先に述べた検査処理は、各プリンタ・システムの性能を保持し、各プリンタ・システムの状況をモニタするネットワーク・プリント・モニタから、各プリンタ・システムの性能及び状況を読み出すよう構成することが可能である。集中的にプリンタ・システムの状況及び性能データを監督するプログラムを設ければ、複数の文書モニタの各々が同機能を保持しなくとも済むという効果が生じる。

【0024】同様のシステムにおいて、プリンタ・システムで実行される処理は、印刷データと印刷コントローラとを含む移動エージェントを受信するステップと、移動エージェントを実行可能状態に変換するステップと、印刷コントローラがそれ自身の設定に従って印刷データを出力するステップと、全ての印刷データを印刷し終えた場合、印刷終了を移動エージェント送信元に通知するステップとを含む。これにより、印刷が無事終了したことを文書モニタに知らせることができる。また、印刷コントローラはそれ自身の設定に従い印刷データをプリンタに出力することができる。なお、印刷終了を移動エージェントからのメッセージで知らせても、移動エージェント自身が文書モニタに戻ることににより通知してもよい。

【0025】プリンタ・システムにエラーが発生した場合、当該エラーの内容を印刷コントローラが検査し、エラーの内容が所定の条件を満たしている場合には、エラーの内容及び未印刷データを含む移動エージェントを、その送信元に送信するように構成する場合もある。場合によっては、エラー内容及び未印刷データに関する情報（又は印刷済みデータ）をメッセージで通知するようにすることも、障害回復まで移動エージェントを休止するようにすることも考えられる。

【0026】印刷データ及び印刷コントローラを含む第2移動エージェントを受信した場合には、第1の移動エージェントによる印刷処理が終了しているか否かを判断するステップと、終了していない場合、第2移動エージェントを実行不能状態で保管するステップとを含むようにすることも考えられる。同一プリンタ・システム内では複数の印刷処理は並列して処理できないため、プリンタ・モニタに到着した順番で移動エージェントを起動し、それ以外は凍結保存する。その後、第1の移動エージェントが印刷処理を終了して、実行状態でなくなった後に、第2の移動エージェントを実行可能状態に戻すことも考えられる。なお、タイミングとしては、印刷処理終了を実行環境が通知された時点で第2の移動エージェントを実行可能とすることも考えられる。

【0027】以上は、処理のフローとして本願発明を表現したが、これらの処理を実行するモジュールを構成し、それらを含むコンピュータ・システム又はローカル・エリア・ネットワークとして発明を実施することも可能である。また、処理のフローをコンピュータ・プログラムとして実施し、CD-ROMやフロッピー・ディスクなどの記憶媒体又は記憶デバイスに記憶することもある。

【0028】

【発明の実施の形態】図1は、システム全体を示した図である。コンピュータ1(3)、コンピュータ2(5)、コンピュータ3(7)、プリンタ・システム1(9)、プリンタ・システム2(11)、及びプリンタ

## 11

・システム3(13)がネットワーク1に接続されている。このコンピュータ1乃至3は、通常のコンピュータであり、ディスプレイやキーボード等の入出力装置を含み、CPUやメイン・メモリを有しており、文書等を作成するアプリケーション・プログラムを実行することができる。コンピュータ1乃至3は、二次記憶装置としてHDDやFDD、CD-ROMドライブ、DVDドライブ等を備えることも可能である。また、プリンタ・システム1乃至3は、コンピュータとプリンタの組み合わせや、ネットワーク対応でコンピュータが内蔵されたプリンタでもよい。なお、プリンタ・システム1乃至3は以下で述べるような印刷ジョブ・エージェントの実行環境(プリンタ・モニタ)を提供できるような能力が必要とされる。エージェントの実行環境とは、移動エージェントが活動するために必要な計算機資源及びサービスを提供するプログラムであり、エージェントの生成・停止・移動等を制御する機構及びエージェント自身やその管理情報などを状態として保持するものである。

【0029】図1には、本発明で用いられる文書モニタ1(15a)及び文書モニタ2(15b)、ネットワーク・プリント・モニタ17、プリンタ・モニタ1(19a)及びプリンタ・モニタ2(19b)及びプリンタ・モニタ3(19c)が示されている。これらは、位置関係を明らかにするために示したものであって、実際は、それぞれのシステムの中に存在するものである。すなわち、コンピュータ1(3)には文書モニタ1(15a)が、コンピュータ2(5)には文書モニタ2(15b)が、コンピュータ3(7)にはネットワーク・プリント・モニタ17が、プリンタ・システム1(9)にはプリンタ・モニタ1(19a)が、プリンタ・システム2(11)にはプリンタ・モニタ2(19b)が、プリンタ・システム3(13)がプリンタ・モニタ3(19c)が存在している。コンピュータ1乃至3における文書モニタの配置は任意である。また、コンピュータ1又は2に、又はすべてのコンピュータに、ネットワーク・プリント・モニタを設けてもよいし、コンピュータ3にも文書モニタが存在するようにしてもよい。

【0030】図1のシステムにおいては、コンピュータ1又は2で作成された印刷データは、文書モニタが印刷すべきプリンタ・システムを決定し、印刷ジョブ・エージェントとして、プリンタ・システムに送信される。プリンタ・システムに設けられたプリンタ・モニタにおいて、印刷ジョブ・エージェントは活動化され、実行すべき印刷ジョブを実行する。全ての印刷が終了すれば、当該印刷ジョブ・エージェントの送信元である文書モニタに印刷終了の通知がなされる。全ての印刷ジョブ・エージェントから印刷終了の通知を受信すると、文書モニタはユーザに対し、どのプリンタにどのページが印刷されたか等の情報を提示する。

【0031】では、図1におけるコンピュータ1内の構

## 12

成を図2に示す。コンピュータ1(3)は、ネットワーク1に接続している。このネットワーク1との接続のため通信モジュール27を含んでいる。さらに、文書等を作成する文書作成アプリケーション・プログラム21が実行されている。また、印刷用のデバイス・ドライバ23が存在している。通常のコンピュータ1(3)内はこのような構成であるが、ユーザが印刷を命じると、印刷する文書データがデバイス・ドライバ23に渡され、さらに文書モニタ15aを活動化する。デバイス・ドライバ23は、文書データを印刷データに変換し、これを活動化した文書モニタ15aに渡す。さらに、デバイス・ドライバ23は、文書データを印刷するに際しユーザが指定したパラメータ(例えば、カラー印刷の指定や、高速印刷又は品質優先の指定、部数や出力形態)を印刷制御データとして文書モニタ15aに出力する。文書モニタ15aは、後述するネットワーク・プリント・モニタ17からプリンタ・システムに関する情報を取得し、印刷制御データと合わせて、プリンタ・システムに対する最適な印刷データの割り当てを決定する。この処理については後に述べる。そして、割り当てが決定されると、割り当てのあったプリンタ・システムごとに印刷ジョブ・エージェント25a乃至25cを生成する。図2では印刷ジョブ・エージェントは3つしか示されていないが、この個数は文書モニタ15aの割り当て処理によって決定される。この印刷ジョブ・エージェント1乃至3(25a乃至25c)は、通信モジュール27を介して宛先のプリンタ・システムに送信される。

【0032】文書モニタ15aは、1つのコンピュータ・システムにおいて1つ設けるようにしてもよいし、1つの印刷命令に1つの文書モニタを生成するようにすることも可能である。1つのコンピュータ・システムに1つの文書モニタを設けるようにする場合には、文書モニタ内で印刷命令ごとに管理が必要となる。以上の構成は、コンピュータ2(7)でも同様である。

【0033】図3は、プリンタ・システム1(9)内の構成を示す図である。プリンタ・システム1(9)は通信モジュール29を介してネットワーク1に接続している。プリンタ・システム1(9)内にはプリンタ・モニタ1(19a)と、プリンタ9aが存在している。また、プリンタ・モニタ1(19a)は、印刷ジョブ・エージェントの実行環境であり、内部で印刷ジョブ・エージェント25dを実行している。さらに、プリンタ・モニタ1(19a)は、エージェント管理モジュール33及びモニタ・モジュール31を含んでいる。印刷ジョブ・エージェント25dは、ネットワーク1及び通信モジュール29を介してプリンタ・モニタ1(19a)に到達する。エージェント管理モジュール33は、受信した印刷ジョブ・エージェント25dを実行可能な形態に変換する処理、その逆で印刷ジョブ・エージェントを凍結する処理、複数の印刷ジョブ・エージェントを受信した

場合には1つの印刷ジョブ・エージェント以外は活動化せずにキューに保管し、可能な時点で取り出す処理等を実行する。また、モニタ・モジュール31は、プリンタ9aを監視し、その監視結果を印刷ジョブ・エージェント25d及びネットワーク・プリント・モニタ19に通知する。印刷ジョブ・エージェント25dは、印刷コントローラを有しており、この印刷コントローラから印刷データがプリンタ9aに出力され、実際に文書が印刷される。他のプリンタ・システムもこれと同様の構成である。

【0034】図4に印刷ジョブ・エージェントのブロック図を示す。印刷ジョブ・エージェントは、印刷データ35及び印刷コントローラを有している。この印刷コントローラは、印刷制御モジュール37及びエラー・ハンドラ39及びプリンタ・モニタ41を含む。印刷制御モジュール37は、プリンタへの印刷データの出力制御が主な機能であり、印刷が無事に成功した場合のメッセージ送信等も実行する。エラー・ハンドラ39は、エラーが発生した場合に起動される。プリンタ・モニタ41はプリンタ・モニタ内のモニタ・モジュール31を介してプリンタの状態を監視し、場合によってはエラー・ハンドラ39に情報を通知する。

【0035】図5にコンピュータ3(7)に設けられるネットワーク・プリント・モニタ17のブロック図を示す。コンピュータ3(7)はネットワーク1に接続されているので、通信モジュールを有している。ネットワーク・プリント・モニタ17は、この通信モジュールを介して他のコンピュータからのプリンタ情報検索命令に回答する検索モジュール45と、プリンタ情報43、プリンタ情報を格納したテーブルのテーブル管理モジュール47を含む。先に述べたように、プリンタ・モニタ19aはモニタ・モジュール31を含んでおり、このモニタ・モジュール31はネットワーク・プリント・モニタ19にプリンタの現在の状態に関する情報を通知する。この情報を受信するのはテーブル管理モジュール47であり、このテーブル管理モジュール47がプリンタ情報を更新する。新たなプリンタ・システムがネットワーク1に接続された場合や既存のプリンタ・システムがネットワーク1から取り除かれた場合、テーブル管理モジュール47に通知すると、このプリンタ情報43を更新する。テーブル管理モジュール47が自発的にネットワーク内のプリンタ・システムを定期的にモニタするようにすることも可能である。また、性能等のデータをプリンタ情報43に含ませることにより、現在の処理状況からターンアラウンド・タイム等を予想する処理を実施するようにしてもよい。

【0036】プリンタ情報43は、図5に示したように、プリンタの動作状況(ダウンしていないかどうか)、カラー出力の可否、印刷品質(印字品質の指定)、1ページあたりの出力時間、現在予約されている印刷文書

の分量等を含む。

【0037】図6にコンピュータ1(3)の処理のフローを示す。最初に、プリンタのデバイス・ドライバ23が文書モニタ15aを活動化する(ステップ110)。そして、文書モニタ15aは、印刷制御データを受信し、ネットワーク・プリント・モニタ19からの情報を取得して、印刷データの割り当てを決定する。この割り当てに基づき、文書モニタ15aは、印刷ジョブ・エージェントを生成する(ステップ120)。そして、生成した印刷ジョブ・エージェントを送信する(ステップ130)。

【0038】生成した印刷ジョブ・エージェントはプリンタ・モニタにて処理を進め、全ての印刷データを印刷し終わると印刷ジョブ・エージェントは送信元の文書モニタに印刷終了を通知する。文書モニタ15aは全ての印刷ジョブ・エージェントから印刷終了の通知を受信したか否か判断する(ステップ140)。もし、印刷終了の通知をすべての印刷ジョブ・エージェントから受信した場合には、ユーザに印刷レポートを出力する(ステップ170)。この印刷レポートは、電子メールの形態であっても、デバイス・ドライバ23によってディスプレイ上に表示するような形態であってもよい。例えば、

プリンタ1 ページ1乃至5

プリンタ2 ページ6

プリント3 ページ7乃至9

等といったデータをユーザに提示する。ユーザはこの情報を元にプリントアウトした紙を取りに行く。

【0039】もし、全ての印刷ジョブ・エージェントから印刷終了を受信していない場合、未印刷データありの通知を受信したか否か判断する(ステップ150)。受信していない場合にはステップ140に戻る。一方、未印刷データありの通知を受け取った場合、例えば、未印刷データを含む印刷ジョブ・エージェント自身が戻ってきた場合、又は未印刷データが存在していることを示すメッセージを受け取った場合、未印刷データを用意し、ステップ120に戻って、この未印刷データに対して印刷ジョブ・エージェントを生成する(ステップ160)。

このような繰り返して、最終的に全ての印刷データがプリントアウトされる。なお、未印刷データの用意は、印刷ジョブ・エージェントが戻ってきた場合にはそれから抽出し、そうでなければ保持している印刷データから抽出する。

【0040】次に、どのように文書モニタが印刷ジョブ・エージェントを生成するかについて図7を用いて説明する。最初に、印刷データ及び印刷制御データをデバイス・ドライバから受信する(ステップ210)。そして、ネットワーク・プリント・モニタにアクセスし、プリンタ情報43を取得する(ステップ220)。プリンタ情報43は図5に示したようなデータである。続いて印刷制御データを検査する(ステップ230)。印刷制

10

20

30

40

50

御データは、印刷要求仕様とページの出力順序等のデータである。印刷要求仕様とは、例えば「高速印刷」「高品質印刷」「カラー印刷」といったユーザの印刷要求のタイプを示す。また、出力順序指定とは、例えば、複数部数印刷する場合に、同一ページをまとめて印刷するか、同一ページをページ順に印刷するか等の選択を示す。

【0041】この印刷制御データとネットワーク・プリント・モニタからの情報から、割り当てポリシーを決定する。すなわち、例えば、カラー印刷を指定された場合には、プリンタ情報43を用いて、カラーを印刷できるプリンタを識別し、カラー印刷の必要なページを割り当てる。また、高速印刷を指定された場合には、プリンタ情報を用いて、よりターンアラウンド・タイムの短いプリンタを割り当ての中心とする。但し、1つのプリンタに集中させると全体としては時間がかかり過ぎることもあるので、1ページ割り当てを実施するごとにターンアラウンド・タイムを見積もるなどの処理を実施するとよい。さらに、高品質印刷が指定された場合には、プリンタ情報を用いて、所定以上の高品質印刷が可能なプリン

\*タを割り当ての中心とする。そして、各ページごとに印刷すべきプリンタを決定する(ステップ240)。

【0042】印刷すべきページが割り当てられたプリンタごとに、印刷コントローラを作成し、印刷ジョブ・エージェントを生成する(ステップ250)。エージェントの作成は、印刷コントローラに対応するプログラム・コードを用意しておき、これに出力順序など印刷制御モジュールに関する設定情報及び印刷データをセットすることにより行う。

10 【0043】現在、図5に示したプリンタ情報43のようなシステムで、4ページ目だけがカラーの合計8枚の文書を印刷するような状況で、「カラー印刷」及び「高速印刷」を指定した印刷要求があったとする。このとき、「カラー印刷」の指定により4ページ目は必然的にPrinter2に割り当てられる。そこで、4ページ目の印刷を行う印刷ジョブ・エージェントを生成する。ここで、条件が以下のように変わる(Printer2が変化)。それとともに最低60秒の出力時間が必要であることが判明する。

【表1】

Printer1 Attribute: B&W Speed: 5 sheets/min in process: 0 sheets

Printer2 Attribute: Color Speed: 1 sheets/min in process: 1 sheets

Printer3 Attribute: B&W Speed: 4 sheets/min in process: 2 sheets

【0044】ここでプリンタごとにn枚の出力を行う場合※ ※合の経過時間は以下の式で算出できる。

Printer1  $T1 = (60/5) * n1 = 12 * n1$

Printer2  $T2 = (60/1) * n2 + (60/1) * 1 = 60 * n2 + 60$

Printer3  $T3 = (60/4) * n3 + (60/4) * 2 = 15 * n3 + 30$

いずれも1次関数として表現される。もっとも単純な決定の手順としては、同一ページをm枚まとめて出力するとしてm枚ごとに3つのプリンタのうちどれで出力するのが全体の最短時間になるのかをチェックしていく方法がある。今回は1枚ごとに出力するため、以下のようになる。

【表2】

出力を Printer1 にした場合には  $T1= 12, T2= 60, T3=30$   $T=\max(12, 60, 30)=60$

出力を Printer2 にした場合には  $T1= 0, T2=120, T3=30$   $T=\max(0, 120, 30)=120$

出力を Printer3 にした場合には  $T1= 0, T2= 60, T3=45$   $T=\max(0, 60, 45)=60$

(T は全体で印刷にかかる時間)

【0045】したがって Printer 1 または Printer3 に出力を行うことが妥当となる。Printer1, Printer3 のうち高速のものを選り、Printer 1 に一枚の出力を割り当てるものとする。同様に二枚目は、と算出される。

【表3】

出力を Printer1 にした場合には  $T1= 24, T2= 60, T3=30$   $T=\max(24, 60, 30)=60$

出力を Printer2 にした場合には  $T1= 0, T2=120, T3=★50$

☆30  $T=\max(0, 120, 30)=120$

30 出力を Printer3 にした場合には  $T1= 0, T2= 60, T3=45$   $T=\max(0, 60, 45)=60$

【0046】よってPrinter 1 に合計2枚が割り当てられることになる。このようにして、n枚の出力の割り当ては

【表4】

n (Printer1, Printer2, Printer3)

1 ( 1, 0, 0)

2 ( 2, 0, 0)

3 ( 3, 0, 0)

4 ( 4, 0, 0)

5 ( 5, 0, 0)

6 ( 5, 0, 1)

7 ( 5, 0, 2)

と変化する。最終的にはPrinter1 へは ページ 1-3 及び ページ 5-6 の 印刷を割り当て、印刷ジョブ・エージェントを生成する。Printer3 へは ページ 7-8 の印刷を割り当て、印刷ジョブ・エージェントを生成する。

【0047】では、図8を用いてプリンタ・システムにおける処理フローを説明する。プリンタ・システムのプリンタ・モニタは、印刷ジョブ・エージェントを受信す

る(ステップ310)。次に、エージェント管理モジュール33は、既に実行中の印刷ジョブ・エージェントが存在するか否かを判断する(ステップ320)。もし、存在する場合には、今受信した印刷ジョブ・エージェントを活動化することなく、バッファに格納する(ステップ330)。バッファは、メインメモリ上に用意してもよいし、二次記憶装置内に用意してもよい。このバッファは基本的にはFIFOであり、到着順に取り出す。但し、印刷ジョブ・エージェントに優先度を設けて、優先度順にバッファから取り出すような機構を設けることも可能である。

【0048】一方、実行中の印刷ジョブ・エージェントが存在しない場合には、エージェント管理モジュール33は当該印刷ジョブ・エージェントを実行可能状態に変換する(ステップ340)。そして、印刷ジョブ・エージェントは自身のプリンタ・モニタ41がプリンタの状況を検査し、エラーが発生していないか判断する(ステップ350)。もし、エラーが発生しているようであれば、エラー・ハンドラ39を起動する(ステップ380)。もし、エラーが発生しなければ、印刷データをプリンタ9aに出力する(ステップ360)。そして、印刷が終了すれば(ステップ370)、送信元に印刷終了を通知する(ステップ390)。この通知は、メッセージを送信するようにすることも、印刷ジョブ・エージェントを返信することも可能である。ここでは、メッセージを送信することにして、印刷終了とともに印刷ジョブ・エージェントを消滅させる(ステップ400)。印刷ジョブ・エージェント内のプリンタ・モニタは、印刷終了までにエラーが発生しないか常時プリンタを観察する。エラーが発生すれば、エラー・ハンドラ39を起動する。

【0049】図9にエラー・ハンドラ39の処理の一例を示す。エラー・ハンドラ39は起動されると、印刷ジョブ・エージェント内のプリンタ・モニタ41からのエラー通知からそのエラー内容を検査する(ステップ510)。ここでは、エラー内容を3つのケースに分けて、ケースごとに処理を変えている。ケース1は、例えばプリンタの電源の立ち上げ待ち状態のような場合を想定し、所定時間待つことにより解消されるものである。よって、エラー・ハンドラ39も待ち状態になり(ステップ520)、所定時間後図8のステップ350に戻る(ステップ530)。

【0050】ケース2は、例えばユーザが直ぐに対処した方がよいようなエラーが発生した場合には、送信元にエラー・メッセージを送信し(ステップ540)、文書モニタ15及びデバイス・ドライバ23を介してユーザに警告を発するようにするものである。また、ケース2とケース3は通常のエラーでどちらかの処置を実施するように設定することも可能であり、エラー・メッセージを送信元に返送することにより、送信元の文書モニタが

未印刷データを用意してそれに対するジョブの割り当てを再度行うようにしてもよい。この時エラー・メッセージには、エラー内容と、未印刷データに関する情報を含める必要がある。さらに、ケース3は、他のプリンタ・システムにて印刷をした方がよい場合であって、エラー内容を格納し(ステップ550)、未印刷データの準備をする(ステップ560)。ここでは、既に印刷されている印刷データについては破棄する。そして、印刷ジョブ・エージェントの凍結処理及び送信元への返送をエージェント管理モジュール33に命ずる(ステップ570)。エージェント管理モジュール33は、命令に従って動作する。凍結処理及び実行可能状態への変換は、「Pickling State in the Java Systems」(The 2nd USENIX Conference on Object-Oriented Technologies, 1996)に実装方法の一例が述べられている。

【0051】以上、本発明の一実施例を示したが、本発明はこれに限定されるものではない。例えば、上の説明では、プリンタ・システムにエラーが発生した場合、印刷ジョブ・エージェントが送信元の文書モニタに戻るような例を示したが、エラー・メッセージにより文書モニタにエラーを通知し、それにより文書モニタが再割り当て処理を実施して、メッセージにより次の移動先プリンタ・モニタを指定するような構成にすることも可能である。また、エラーの発生したプリンタ・システムのプリンタ・モニタ上で印刷ジョブを文書モニタからのメッセージに従って分割するようなことも考えられる。また、図2乃至図5に示した文書モニタ、プリンタ・モニタ、印刷ジョブ・エージェント、及びネットワーク・プリンタ・モニタ内のモジュール分けは任意であって、このようなモジュール分けに限定されるわけではなく、上で述べたような機能を有するように、モジュールを分離合併可能である。

【0052】

【効果】ネットワークに複数のプリンタが接続されている環境で、あるプリンタに障害が発生しても、自動的に別のプリンタを選択しそのプリンタで印刷を実行できるようにする方法を提供することができた。

【0053】また、ネットワークに複数のプリンタが接続されている環境で、複数のプリンタを組み合わせる並列印刷可能とする方法を提供することもできた。

【0054】さらに、移動エージェントの技術を用いて、印刷処理を実施する方法を提供することもできた。

【図面の簡単な説明】

【図1】本発明の全体構成を示す図である。

【図2】コンピュータ1における構成を示すブロック図である。

【図3】プリンタ・システム1における構成を示すブロック図である。

19

【図4】印刷ジョブ・エージェントのブロック図である。

【図5】ネットワーク・プリント・モニタのブロック図である。

【図6】コンピュータ1における処理フローを示す図である。

【図7】文書モニタの印刷データの割り振りに関する処理フローを示す図である。

【図8】プリンタ・システム1における処理フローを示す図である。

【図9】エラー・ハンドラの処理フローを示す図である。

【符号の説明】

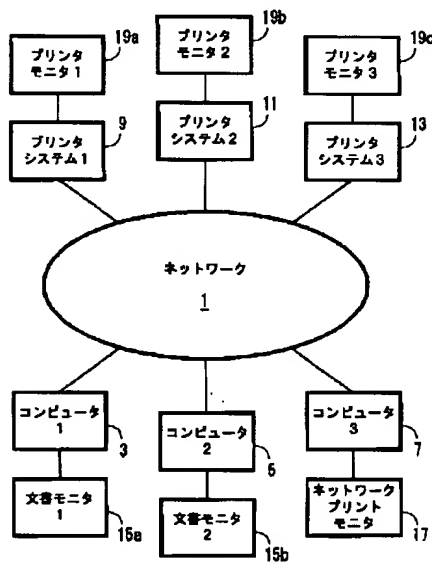
- 1 ネットワーク 3 コンピュータ1  
5 コンピュータ2 7 コンピュータ3  
9 プリンタ・システム1  
9a プリンタ  
11 プリンタ・システム2  
13 プリンタ・システム3  
15a 文書モニタ1

20

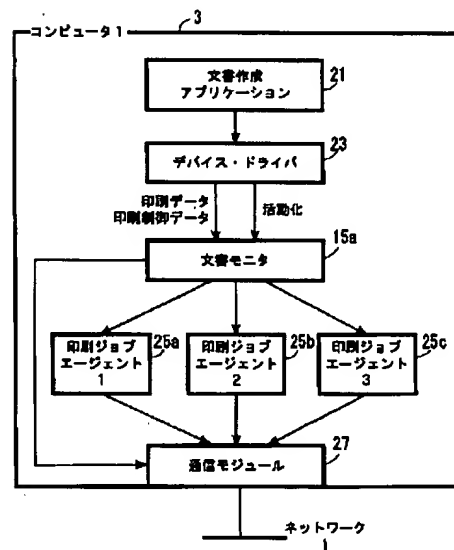
- 15b 文書モニタ2  
17 ネットワーク・プリント・モニタ  
19a プリンタ・モニタ1  
19b プリンタ・モニタ2  
19c プリンタ・モニタ3  
21 文書作成アプリケーション  
23 デバイス・ドライバ  
25a乃至d 印刷ジョブ・エージェント  
27 通信モジュール  
29 通信モジュール  
31 モニタ・モジュール  
33 エージェント管理モジュール  
35 印刷データ  
37 印刷制御モジュール  
39 エラー・ハンドラ  
41 プリンタ・モニタ  
43 プリンタ情報  
45 検索モジュール  
47 テーブル管理モジュール

20

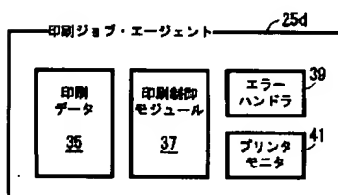
【図1】



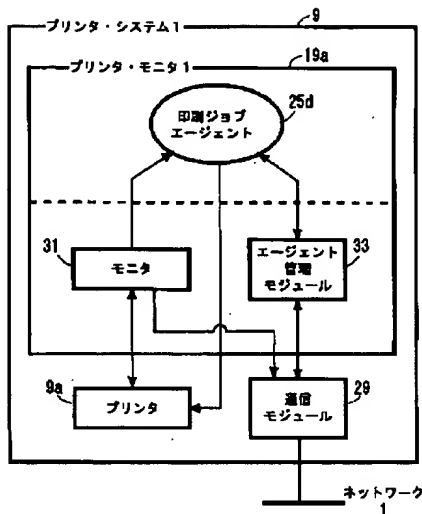
【図2】



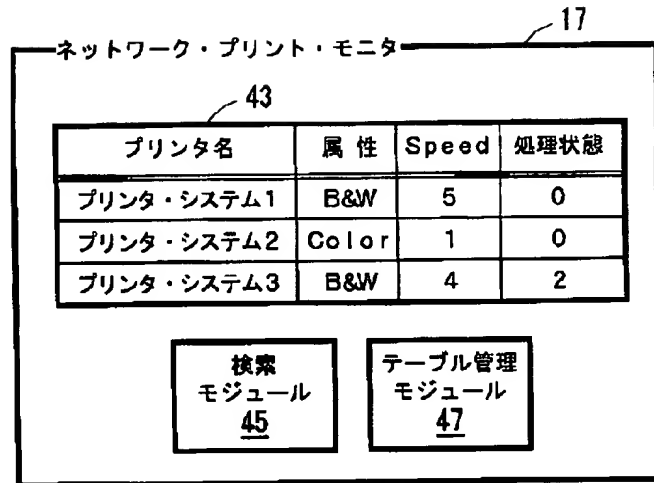
【図4】



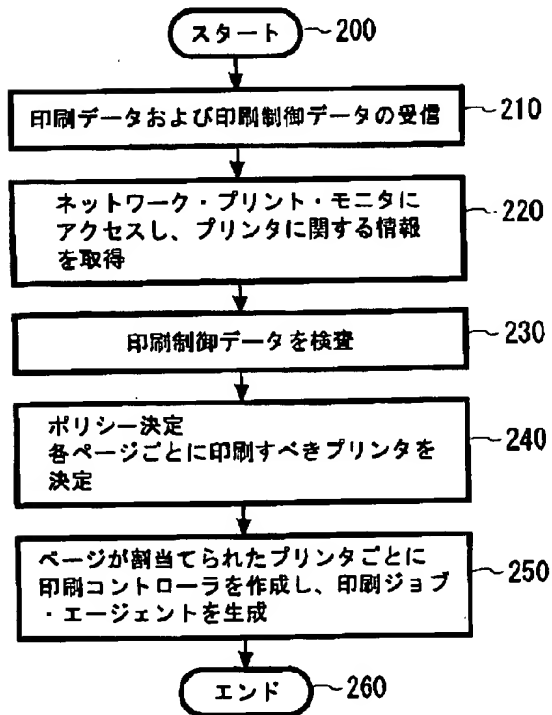
【図3】



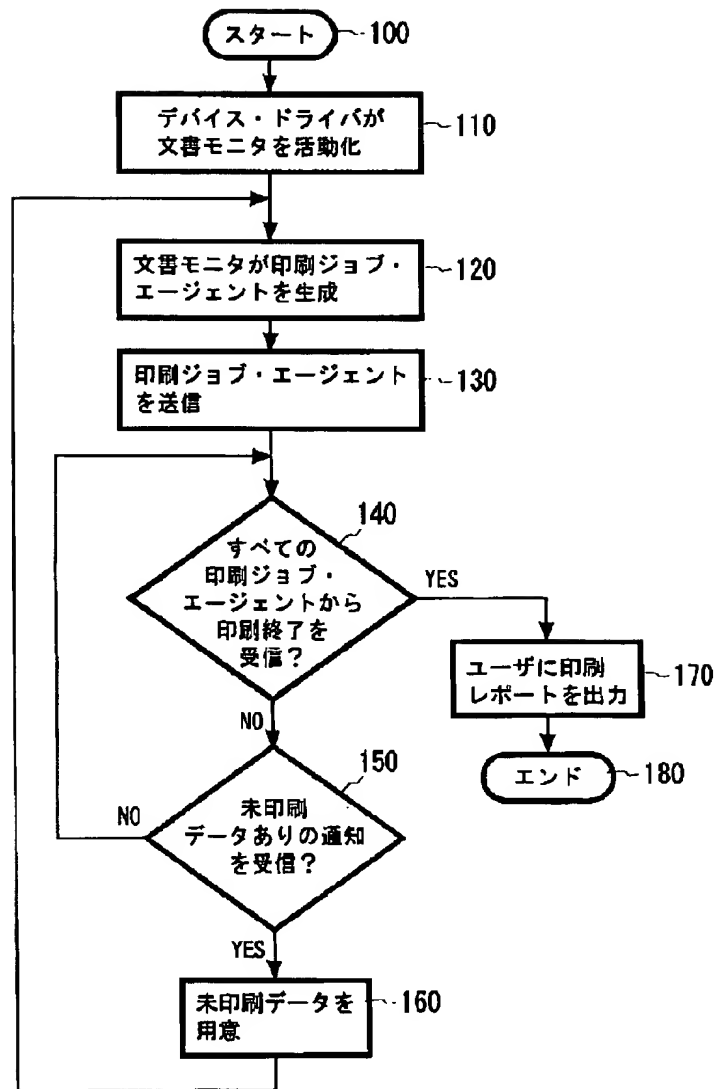
【図5】



【図7】

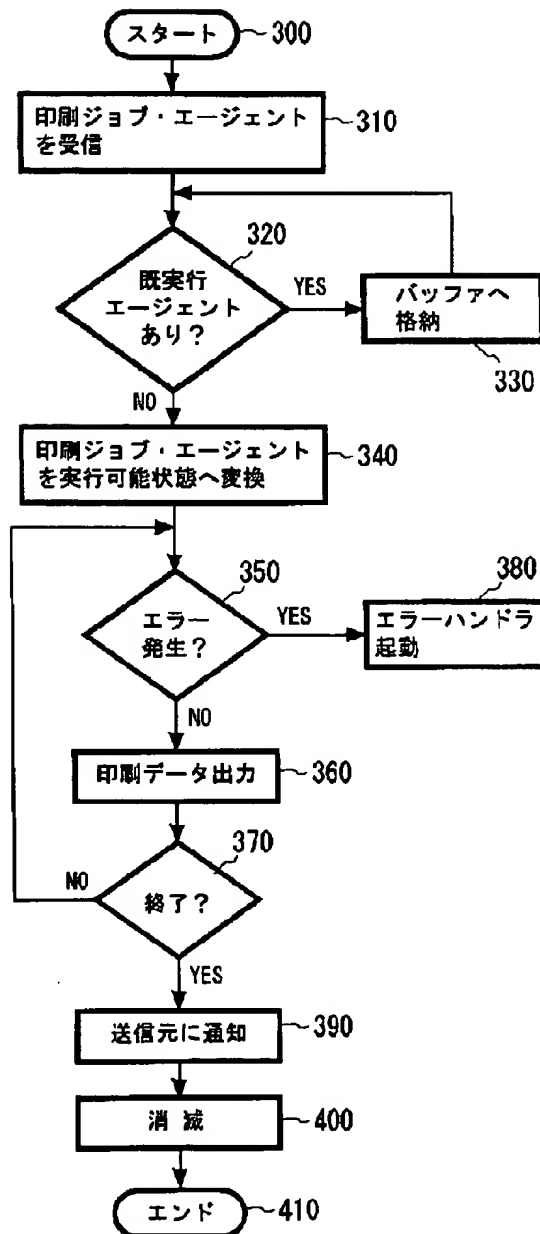


【図6】

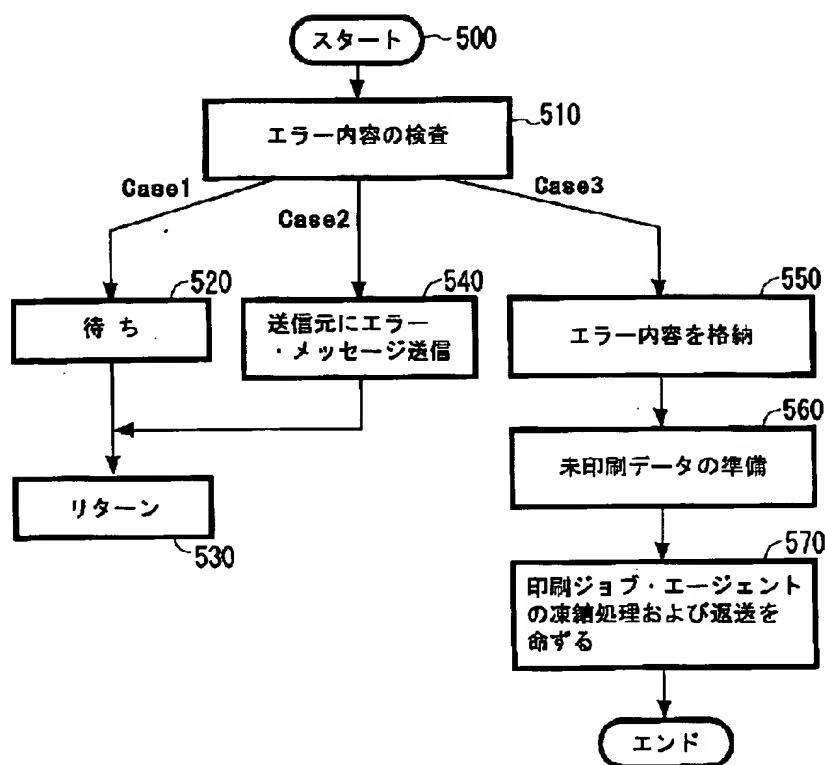




【図8】



【図9】



## 【手続補正書】

【提出日】平成10年6月11日

## 【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】請求項26

【補正方法】変更

## 【補正内容】

【請求項26】印刷用デバイス・ドライバを格納した記憶デバイスであって、前記印刷用デバイス・ドライバは、印刷データと、当該印刷データの印刷制御データとを受け取るステップと、複数のプリンタ・システムに関する情報を取得する検査ステップと、前記複数のプリンタ・システムに関する情報と前記印刷制御データとを参照して、前記印刷データをどのプリンタ・システムにどのように割り振るか決定し、決定された割り振りに従い前記印刷データを分割する分割ステップと、分割された印刷データの各々について、当該分割された印刷データと印刷コントローラを含む移動エージェントを作成するステップとを実行するプログラムを起動するステップと、前記プログラムに印刷データ及び印刷制御データを出力するステップとを含む、記憶デバイス。

## 【手続補正2】

【補正対象書類名】明細書

【補正対象項目名】0015

【補正方法】変更

## 【補正内容】

【0015】文書モニタは、文書を印刷するときに生成される印刷データをもったプロセスである。特に、文書を複数に分割して印刷する場合に有効なしくみである。このプロセスでは印刷することを要求されたページをどのように分割し、どのプリンタに作業を割り当てたか、実行中の印刷作業がどのような状態にあるのか、どのプリンタがどれぐらいの待ち時間で利用でき、またカラー印刷ができるか、どれぐらいの処理能力があるかなどといった印刷のための資源の管理を行う。文書モニタは印刷するためのデータの情報ならびに、プリンタ単位で生成した印刷ジョブの表を持っている。同時に、印刷が終了したとき、どのページがどのプリンタで出力されているかなどを管理し報告する機能をもつ。このプロセスは必要に応じて生成され文書印刷とそのレポートを作成したあとは消滅するようにすることも可能である。

フロントページの続き

(72)発明者 小坂 一也  
神奈川県大和市下鶴間1623番地14 日本ア  
イ・ビー・エム株式会社 東京基礎研究所  
内